

What is claimed is:

1. A Montgomery modular multiplier which calculates a value congruent to " ABR^{-1} " (mod M), where A and B are input n -bit numbers, R^{-1} is an inverse number of R modular-multiplied for "mod M ", and M is a modulus, the Montgomery modular multiplier comprising:

an A-register storing a bit value a_i (where i is an integer ranging from 0 to $n-1$) of the number A , which is smaller than the modulus M ;

a B-register storing a bit value b_i of the number B , which is smaller than the modulus M ;

an M-register storing a bit value m_i of the modulus M , which is an odd number;

a b_iA calculation logic circuit multiplying the number A by b_i to obtain b_iA ;

a q_i calculation logic circuit solving a Boolean logic equation " $s_0 \text{ XOR } c_0 \text{ XOR } (b_i \text{ AND } a_0)$ ", where s_0 is the least significant bit (LSB) of a sum S , c_0 is the LSB of a carry C , b_i is the bit value of the number B , and a_0 is the LSB of the number A , to obtain a bit value q_i ;

a q_iM calculation logic circuit multiplying M by q_i to obtain q_iM ;

a 4-2 compressor first performing n additions on the carry C , the sum S , the b_iA , and the q_iM to obtain interim values summing the interim values to obtain a result by carry propagation adder in response to a carry propagation adder signal;

an S-register updating a bit value s_i of the sum S and storing the updated bit value; and

a C-register updating a bit value c_i of the carry C and storing the updated bit value.

2. The Montgomery modular multiplier of claim 1, wherein the 4-2 compressor comprises:

a first full adder unit, summing a bit value b_ia_i of b_iA , a bit value s_{i+1} of the sum S , and the bit value c_i of the carry C to obtain a carry cA_i and a sum sA_i ;

a MUX unit selectively outputting either a bit value $q_i m_i$ of $q_i M$, the carry cA_{i-1} , and sA_i or the bit value s_{i+1} of S , the bit value c_i of the carry C , and the bit value c_{i-1} of the C , in response to the carry propagation adder signal;

a second full adder unit performing n additions on the bit value $q_i m_i$ of $q_i M$, cA_{i-1} , and sA_i to calculate interim bit values s_i and c_i of S and C , when the carry propagation adder signal is in an inactive state, and then summing the bit value s_{i+1} of S , the bit value c_i of C , and the bit value c_{i-1} of C to obtain final results of the sum S and carry C , when the carry propagation adder signal is in an active state.

3. The Montgomery modular multiplier of claim 1, wherein the carry save adder structure is a 4-input 2-output structure, in which the first and second full adder units operate when the carry propagation adder signal is in an inactive state.

4. The Montgomery modular multiplier of claim 1, wherein the carry propagation adder structure is a 3-input 2-output structure, in which only the second full adder unit operates when the carry propagation adder signal is in an active state.

5. The Montgomery modular multiplier of claim 2, wherein the LSB of the carry cA_{i-1} and the LSB of the carry c_{i-1} are in a first logic state.

6. The Montgomery modular multiplier of claim 2, wherein the Most Significant Bit (MSB) of the sum s_{i+1} is equal to the carry cA_{n-1} at a clock pulse before the carry propagation adder signal is activated.

7. A method of performing a Montgomery modular multiplication in a Montgomery modular multiplier, which includes registers for storing bit values a_i , b_i , m_i , c_i , and s_i (where i denotes an integer in the range of 0 to $n*1$) of a word A , a word B , a modulus M , a carry C , and a sum S , respectively, and calculates a value congruent to " ABR^{-1} " (mod M), where A and B are input n -bit numbers, R^{-1} is an inverse number of R modular-multiplied for "mod M ", and M is a modulus, the method comprising:

receiving the number A, the number B, and the modulus M;
 multiplying the number A by a bit value b_i to obtain each bit of b_iA ;
 solving a Boolean logic equation " $s_0 \text{ XOR } c_0 \text{ XOR } (b_i \text{ AND } a_0)$ ", where s_0 is the least significant bit (LSB) of a sum S, c_0 is the LSB of a carry C, b_i is the bit value of the number B, and a_0 is the LSB of the number A, to obtain a bit value q_i (where i denotes an integer in the range of 0 to $n-1$);
 multiplying the number M by the bit value q_i to obtain each bit of q_iM ;
 performing n additions on the carry C, the sum S, the b_iA , and the q_iM to obtain interim values for each bit of the sum S and the carry C in a carry save adder structure, in response to a carry propagation adder signal; and
 summing the interim values to obtain the final results of the sum S and carry C in a carry propagation adder structure, in response to the carry propagation adder signal.

8. The method of claim 7, wherein the number A is smaller than the modulus M.

9. The method of claim 7, wherein the number B is smaller than the modulus M.

10. The method of claim 7, wherein the modulus M is an odd number.

11. The method of claim 7, wherein the interim and final values of S and the interim and final values of C are calculated by:

summing a bit value b_iA_i of the b_iA , a bit value s_{i+1} of S, and the bit value c_i of C to obtain a carry cA_i and sA_i ;

selectively outputting either a bit value q_iM_i of the q_iM , cA_{i-1} , and sA_i or the bit value s_{i+1} of S, the bit value c_i of C, and a bit value c_{i-1} of C, in response to the carry propagation adder signal;

performing n additions on the bit value q_iM_i of the q_iM , cA_{i-1} , and sA_i to calculate interim bit values s_i and c_i of S and C, when the carry propagation adder signal is in an inactive state; and

summing the bit value s_{i+1} of S, the bit value c_i of C, and the bit value c_{i-1} of C to obtain final results of S and C, when the carry propagation adder signal is in an active state.

12. The method of claim 7, wherein the carry save adder structure is a 4-input 2-output structure, in which the interim values of S and C are obtained from the b_iA and q_iM when the carry propagation adder signal is in an inactive state.

13. The method of claim 7, wherein the carry propagation adder structure is a 3-input 2-output structure, in which the final values of S and C are obtained from the interim values of S and C when the carry propagation adder signal is in an active state.

14. The method of claim 11, wherein the LSB of cA_{i-1} and the LSB of c_{i-1} are in a first logic state.

15. The method of claim 11, wherein the MSB value of s_{i+1} is equal to the bit value cA_{n-1} at a clock before the carry propagation adder signal is activated.

16. A method of performing radix 2^N Montgomery multiplication, where $N \geq 1$, comprising;
receiving a multiplicand (A), a modulus (M), and a multiplier (B);
performing carry save addition on at least four inputs related to the multiplicand, modulus, and multiplier to generate a result in redundant representation; and
performing carry propagation addition to generate a result in normal representation in response to a carry propagation adder signal.

17. A Montgomery multiplier comprising:
a multiplicand (A) register, storing a bit value a_i of the number A;
a modulus (M) register, storing a bit value m_i of the number M;
a multiplier (B) register, storing a bit value b_i of the number B;

a b_iA calculation logic circuit multiplying the number A by a bit value b_i to obtain each bit of b_iA ;

a q_i calculation logic circuit solving a Boolean logic equation " $s_0 \text{ XOR } c_0 \text{ XOR } (b_i \text{ AND } a_0)$ ", where s_0 is the least significant bit (LSB) of a sum S , c_0 is the LSB of a carry C , b_i is the bit value of the number B , and a_0 is the LSB of the number A , to obtain a bit value q_i (where i denotes an integer in the range of 0 to $n*1$);

a q_iM calculation logic circuit multiplying the modulus M by the bit value q_i to obtain each bit of q_iM ; and

a t - s compressor, wherein $t > 3$, and $s > 1$, performing n additions on the carry C , the sum S , the b_iA , and the q_iM to obtain interim values for each bit of the sum S and the carry C in a carry save adder structure and summing the interim values to obtain final results of the S and C in a carry propagation adder structure, in response to a carry propagation adder signal.